接圆回归*

何沧平

(微博, cangping@staff.weibo.com, cphe@lsec.cc.ac.cn)

摘　要

本文提出一个名为接圆回归的点击率预测新方法，尝试替代常用的因子分解机(FM)。接圆回归用超平面拼接出一个封闭凸多面体，圈出正样本，有直观的几何解释，能从任意初始值一次收敛到全局最优解。拟合出来的曲面Lipschitz连续，变化平缓。在人工设计的星环集、双堆集、双月集上，接圆回归的分类准确性、解释性、平滑性全面超过FM。在同量级参数量、计算量的条件下，接圆回归在Avazu集和Criteo集上的AUC超过FM。

关键词： 接圆回归，楔子正则，因子分解机

Polyhedron Regression

He Cangping

(*weibo.com, cangping@staff.weibo.com, cphe@lsec.cc.ac.cn*)

Abstract

This paper proposes a novel method named Polyhedron Regression(PR) for Click-Through-Rate prediction, aiming to take the place of Factorization Machines(FM). PR constructs a convex polyhedra with hyperplanes to separate positive samples from negative samples. PR has intuitionistic geometrical interpretations and a Lipschitz continuous surface, converges to global optimum point from arbitrary initial values. Compared with FM, PR has better classification accuracy, interpretability and surface smoothness on the three artificial datasets. With comparable parameters and computation, PR achieves better AUC than FM on Avazu and Criteo datasets.

Keywords: polyhedron regression, wedge regularization, factorization machines

1. 引言

大量的网站、手机应用采用信息流推荐技术，推荐效果直接影响企业收入。推荐过程大致是这样，以微博热门流为例，当用户刷新页面时，云端推荐系统立即从物料库中粗选出一批微博，几十至几百条，然后将这批微博送入排序模型进行打分，再根据各条微博的得分情况挑选出一部分进行曝光。对排序模型的核心要求有2个：速度快、效果好。实际业务中，排序的时间配额在20毫秒以下，时间过长会影响用户体验；效果好就是用户的点击率高。

在实际使用中发现，Wide & Deep [8] 之类的深度学习模型有几个难点：解释性差，排序效果提升有限，消耗的算力却成十倍地增加。在线机器学习中，模型要快速更新，例如30分钟更新一次，这就要求模型训练要快速度完成。而深度学习模型收剑性依赖于参数初值，不保证每次都收敛到全局最优解，快速更新与参数最优难以兼得，服务器成本还很高。

* 2019年4月2日提交.

简单模型因子分解机(Factorization Machines, FM)广泛使用，它解释性好，计算量较小。但FM也有一些缺点：捕获交叉特征是人类视角的解释，不是严格的数学解释；拟合的曲面变化剧烈（见图1)，与事物平缓变化的经验不符；收敛性依赖于初值，容易陷入局部最优解。

本文提出一种叫做*接圆回归*的排序方法，它保留了FM计算量小的优点，又克服了FM的缺点。接圆回归有直观清晰的几何解释：用多个折面拼接出一个封闭凸多面体，多面体内是正样本，多面体外是负样本。计算量小，且与折面数量成正比，可以通过指定折面数来灵活调整。拟合的曲线变化平缓，Lipschitz连续。从任意初值出发训练，都能一次收敛到全局最优解。

接圆回归有望接替逻辑回归 [1]和FM，组件楔子正则也可以应用到各种深度学习模型当中。

本文后续内容这样组织。第2节相关工作给出当前流行的排序模型，第3节给出接圆回归的公式，第4节介绍至关重要楔子正则，第5节给出偏导数的推导过程，第6节给出小批量计算时的偏导数，第7节给出真实数据集上的实验结果，第8节讨论总结全文。

2. 相关工作

本节只给出目前工业界常用的CTR算法。

逻辑回归 [1](Logistic Regression)，形式简单，计算量小，在推荐系统中广泛应用。MLR [3]改进逻辑回归，提出用分片平面分隔正负样本。因子分解机 [4–6](Factorization Machines, FM)使用交叉特征，实践中常用二阶交叉特征，计算量只比逻辑回归增加了k倍，这里的k是FM的隐向量数量。FFM [7]按域组织交叉特征，隐向量数k远小于FM，但参数量和计算量都成倍增加。

逻辑回归的流行实现库为LIBLINEAR [2]，FM的官方实现库为libFM [5]，FFM的官方实现库为libFFM[1) 。xLearn[2) 是新近出现的算法库，它几乎囊括LIBLINEAR、libFM、libFFM 的全部功能，并且具有更好的性能、易用性和可扩展性。

将神经网络与传统机器学习算法结合，同时捕获高阶特征和低阶特征，得到了一些新算法。Wide & Deep [8]给合神经网络与逻辑回归；deepFM [9, 10]结合神经网络与FM；AFM [12]用神经网络来优化FM的隐参数；FNN [11]先用FM学习到的隐向量作为神经网络的输入，再由神经网络完成最终学习；Deep & Cross [13]不需要特征工程就能获得高阶的交叉特征，比FM系列模型有更高的计算效率；xDeepFM [14]自动学习显式的高阶特征交互；DIN [15]设计了一个attention结构，引入用户的历史行为。

3. 接圆回归

给定数据集$D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$，$d$维列向量$x^{(k)} = (x_1^{(k)}; x_2^{(k)}; \dots; x_d^{(k)})$，标量$y^{(k)} \in \{0, 1\}$。当$y^{(k)} = 0$时，称$x^{(k)}$是负样本；当$y^{(k)} = 1$时，称$x^{(k)}$是正样本。$D$中正、负样本的数量分别记为$n_1$和$n_0$，显然有$n_0 + n_1 = n$。二分类问题的目标是从数据集$D$中学习到一个模型，然后用这个模型预测任意的样本$x$所属的类别。

对$\forall (x, y) \in D$和任意给定的正整数m，定义接圆回归

$$z_i = \min\left(w_i^T x + b_i, -w_i^T x + \tilde{b}_i\right), \quad i = 1, 2, ..., m, \tag{3.1}$$

$$\bar{z} = \frac{1}{m}\sum_{i=1}^{m} z_i, \tag{3.2}$$

$$a = \sigma(\bar{z}), \tag{3.3}$$

这里的$w_i = (w_{1i}, w_{2i}, \dots, w_{di})^T$是$d$维列向量，$b_i$和$\tilde{b}_i$是实数，Sigmoid函数$\sigma(\bar{z}) = \frac{1}{1+e^{-\bar{z}}}$。

1) https://www.csie.ntu.edu.tw/~cjlin/libffm/
2) https://github.com/aksnzhy/xlearn

为了简洁，也为了提高在计算机上的运算速度，记矩阵$W = [w_1, w_2, \ldots, w_m]$，列向量$b = (b_1, b_2, \ldots, b_m)^T$，列向量$\tilde{b} = (\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_m)^T$，列向量$z = (z_1, z_2, \ldots, z_m)^T$, 函数$\text{mean}(z) = \frac{1}{m}\sum_{i=1}^{m} z_i$。将式(3.1)和式(3.2)向量化为

$$\begin{aligned} z &= \min\left(W^T x + b, -W^T x + \tilde{b}\right), & (3.4)\\ \bar{z} &= \text{mean}(z). & (3.5)\end{aligned}$$

样本(x, y)上的损失函数定义为

$$h(x) = \begin{cases} -\frac{n_1}{n_0}\ln(1-a), & 若\ y = 0,\\ -\ln a, & 若\ y = 1.\end{cases} \quad (3.6)$$

数据集D上的损失函数定义为

$$\begin{aligned} H(W, b, \tilde{b}) &= \frac{1}{n}\sum_{(x,y)\in D} h(x) + \frac{\eta_1}{m}\sum_{i=1}^{m}|w_i| + \frac{\eta_2}{m(m-1)}\sum_{i=1}^{m}\sum_{j=i+1}^{m}\text{wed}(\cos\theta_{ij}, \cos\theta_0) \quad (3.7)\\ &= H_1(W, b, \tilde{b}) + \eta_1 H_2(W) + \eta_2 H_3(W)\end{aligned}$$

这里的$H_1(W, b, \tilde{b})$是样本损失，$H_2(W)$是L2正则损失，$H_3(W)$是楔子正则损失，非负实数η_1和η_2为相应的正则数化系数。接圆回归对应的优化问题是

$$\{W^*, b^*, \tilde{b}^*\} = \underset{W\in R^{d\times m}, b\in R^m, \tilde{b}\in R^m}{\arg\min} H(W, b, \tilde{b}). \quad (3.8)$$

4. 楔子正则

式(3.7)中楔子正则的作用是让多个向量均匀分布，不要挤在一起，就像在任意两个向量之间都塞了一个楔子。

对m个d列向量$w_1, w_2, \ldots, w_m$和$\forall i, j = 1, 2, \ldots, m$，向量$w_i$和$w_j$的夹角记为$\theta_{ij}$，则夹角余弦为

$$\cos\theta_{ij} = \frac{w_i^T w_j}{|w_i|\cdot|w_j|},$$

显然有$\cos\theta_{ij} = \cos\theta_{ji}$。定义楔子函数

$$\text{wed}(c, c_0) = \begin{cases} -\ln(1-c^2) + \ln(1-c_0^2), & 若\ |c| \geq |c_0|,\\ 0, & 若\ |c| < |c_0|.\end{cases}$$

这里的$0 \leq c_0 \leq 1$是任意指定的实数。对任意给定的实数$\theta_0 \in [0, \frac{\pi}{2}]$，楔子损失定义为

$$H_3(W) = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m}\text{wed}(\cos\theta_{ij}, \cos\theta_0).$$

为了计算$H_3(W)$的偏导数，先计算$\cos\theta_{ij}$的偏导数。对$\forall k = 1, 2, \ldots, m$，易得

$$\frac{\partial\cos\theta_{ij}}{\partial w_k} = \begin{cases} \frac{w_j}{|w_k|\cdot|w_j|} - \frac{w_k}{|w_k|^2}\cos\theta_{kj}, & 若\ k = i,\\ \frac{w_i}{|w_k|\cdot|w_i|} - \frac{w_k}{|w_k|^2}\cos\theta_{ki}, & 若\ k = j,\\ 0, & 若\ k \neq i 且\ k \neq j.\end{cases}$$

楔子函数的导数为

$$\mathrm{wed}'(c,c_0)=\frac{\mathrm{d}\,\mathrm{wed}(c,c_0)}{\mathrm{d}c}=\begin{cases}\frac{2c}{1-c^2}, & 若\ |c|\ge|c_0|,\\ 0, & 若\ |c|<|c_0|.\end{cases}$$

从而，楔子损失的偏导数为

$$\begin{aligned}\frac{\partial H_3(W)}{\partial w_k} &= \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m}\mathrm{wed}'(\cos\theta_{ij},\cos\theta_0)\frac{\partial\cos\theta_{ij}}{\partial w_k}\\ &= \frac{1}{m(m-1)}\sum_{\substack{i=1\\ i\neq k}}^{m}\mathrm{wed}'(\cos\theta_{ki},\cos\theta_0)\left(\frac{w_i}{|w_k|\cdot|w_i|}-\frac{w_k}{|w_k|^2}\cos\theta_{ki}\right), \qquad (4.1)\\ \frac{\partial H_3(W)}{\partial W} &= \left[\frac{\partial H_3(W)}{\partial w_1},\frac{\partial H_3(W)}{\partial w_2},\cdots,\frac{\partial H_3(W)}{\partial w_m}\right]. \qquad (4.2)\end{aligned}$$

5. 计算偏导数

用随机梯度法(SGD)等迭代方法求解最优化问题(3.8)时，会用到$H(W,b,\tilde{b})$的偏导数，由于min函数的存在，偏导数形式有点复杂，因此本节给出偏导数的推导过程。

令f和$\tilde{f}$均为m维列向量，满足$\tilde{f}=1-f$。记$f=(f_1,f_2,\dots,f_m)^T$，对$\forall i=1,2,\dots,m$，将f_i定义为

$$f_i=\begin{cases}1, & 若\ z_i=w_i^Tx+b_i,\\ 0, & 若\ z_i=-w_i^Tx+\tilde{b}_i\ 且\ z_i\neq w_i^Tx+b_i,\end{cases}$$

简记为$f_i=(w_i^Tx+b_i\le -w_i^Tx+\tilde{b}_i)$。从而式(3.1)可改写为

$$z_i=(w_i^Tx+b_i)f_i+(-w_i^Tx+\tilde{b}_i)\tilde{f}_i,\quad i=1,2,...,m. \qquad (5.1)$$

由式(5.1)(3.2)(3.3)(3.6)易得偏导数

$$\frac{\partial h(x)}{\partial\bar{z}}=\frac{\partial h(x)}{\partial a}\frac{\partial a}{\partial\bar{z}}=\begin{cases}\frac{n_1}{n_0}a, & 若\ y=0,\\ a-1, & 若\ y=1.\end{cases}$$

$$\frac{\partial\bar{z}}{\partial z_i}=\frac{1}{m},\quad \frac{\partial z_i}{\partial w_i}=(f_i-\tilde{f}_i)x,\quad \frac{\partial z_i}{\partial b_i}=f_i,\quad \frac{\partial z_i}{\partial\tilde{b}_i}=\tilde{f}_i$$

经过链式求导，得

$$\frac{\partial h(x)}{\partial b_i}=\frac{\partial h(x)}{\partial\bar{z}}\frac{\partial\bar{z}}{\partial z_i}\frac{\partial z_i}{\partial b_i}=\begin{cases}\frac{n_1}{n_0m}af_i, & 若\ y=0,\\ \frac{1}{m}(a-1)f_i, & 若\ y=1.\end{cases} \qquad (5.2)$$

$$\frac{\partial h(x)}{\partial\tilde{b}_i}=\frac{\partial h(x)}{\partial\bar{z}}\frac{\partial\bar{z}}{\partial z_i}\frac{\partial z_i}{\partial\tilde{b}_i}=\begin{cases}\frac{n_1}{n_0m}a\tilde{f}_i, & 若\ y=0,\\ \frac{1}{m}(a-1)\tilde{f}_i, & 若\ y=1.\end{cases} \qquad (5.3)$$

$$\frac{\partial h(x)}{\partial w_i}=\frac{\partial h(x)}{\partial\bar{z}}\frac{\partial\bar{z}}{\partial z_i}\frac{\partial z_i}{\partial w_i}=\begin{cases}\frac{n_1}{n_0m}a(f_i-\tilde{f}_i)x, & 若\ y=0,\\ \frac{1}{m}(a-1)(f_i-\tilde{f}_i)x, & 若\ y=1.\end{cases} \qquad (5.4)$$

式(5.2)-(5.4)对应的向量形式偏导数分别为

$$\frac{\partial h(x)}{\partial b}=\begin{cases}\frac{n_1}{n_0m}af, & 若\ y=0,\\ \frac{1}{m}(a-1)f, & 若\ y=1.\end{cases} \qquad (5.5)$$

$$\frac{\partial h(x)}{\partial \tilde{b}} = \begin{cases} \frac{n_1}{n_0 m} a\tilde{f}, & \text{若 } y=0, \\ \frac{1}{m}(a-1)\tilde{f}, & \text{若 } y=1. \end{cases} \tag{5.6}$$

$$\frac{\partial h(x)}{\partial W} = \begin{cases} \frac{n_1}{n_0 m} ax(f-\tilde{f})^T, & \text{若 } y=0, \\ \frac{1}{m}(a-1)x(f-\tilde{f})^T, & \text{若 } y=1. \end{cases} \tag{5.7}$$

从而

$$\frac{\partial H_1(W,b,\tilde{b})}{\partial b} = \frac{1}{n}\sum_{(x,y)\in D}\frac{\partial h(x)}{\partial b}, \quad \frac{\partial H_1(W,b,\tilde{b})}{\partial \tilde{b}} = \frac{1}{n}\sum_{(x,y)\in D}\frac{\partial h(x)}{\partial \tilde{b}} \tag{5.8}$$

$$\frac{\partial H_1(W,b,\tilde{b})}{\partial W} = \frac{1}{n}\sum_{(x,y)\in D}\frac{\partial h(x)}{\partial W}. \tag{5.9}$$

$H_2(W)$偏导数容易计算

$$\frac{\partial H_2(W)}{\partial W} = \frac{1}{m}\left[\frac{w_1}{|w_1|}, \frac{w_2}{|w_2|}, \cdots, \frac{w_m}{|w_m|}\right].$$

因此得

$$\begin{aligned} \frac{\partial H(W,b,\tilde{b})}{\partial b} &= \frac{\partial H_1(W,b,\tilde{b})}{\partial b}, \\ \frac{\partial H(W,b,\tilde{b})}{\partial \tilde{b}} &= \frac{\partial H_1(W,b,\tilde{b})}{\partial \tilde{b}}, \\ \frac{\partial H(W,b,\tilde{b})}{\partial W} &= \frac{\partial H_1(W,b,\tilde{b})}{\partial W} + \eta_1\frac{\partial H_2(W,b,\tilde{b})}{\partial W} + \eta_2\frac{\partial H_3(W,b,\tilde{b})}{\partial W}. \end{aligned}$$

6. 小批量计算

在实际应用场景中，训练样本数量n通常很大，在百万以上甚至达到百亿数量级。为了减少计算量，每步只在一小批样本上训练。如果使用式(5.8)(5.9)来计算在一批样本上的偏导数，每个样本都要计算1次，效率不高。为了将一批样本上的偏导数同时计算出来，本节将其转化为更大规模的矩阵运算。

将小批量样本集记为$\hat{D} \subset D$，假设$\hat{D}$中包含$\hat{n}$个样本，正负样本数量分别为$\hat{n}_1$和$\hat{n}_0$。简便起见，$\hat{D}$中的编号，负样本在前，正样本在后，即$\hat{D} = \{x^{(1)}, x^{(2)}, \ldots, x^{(\hat{n}_0)}, x^{(\hat{n}_0+1)}, x^{(\hat{n}_0+2)}, \ldots, x^{(\hat{n}_0+\hat{n}_1)}\}$。令负样本矩阵$X^{[0]} = \left[x^{(1)}, x^{(2)}, \ldots, x^{(\hat{n}_0)}\right]$，大小为$d \times \hat{n}_0$；$B^{[0]} = [b, b, ..., b]$，大小为$m \times \hat{n}_0$；$\tilde{B}^{[0]} = [\tilde{b}, \tilde{b}, ..., \tilde{b}]$，大小为$m \times \hat{n}_0$。令正样本矩阵$X^{[1]} = \left[x^{(\hat{n}_0+1)}, x^{(\hat{n}_0+2)}, \ldots, x^{(\hat{n}_0+\hat{n}_1)}\right]$，大小为$d \times \hat{n}_1$；$B^{[1]} = [b, b, ..., b]$，大小为$m \times \hat{n}_1$；$\tilde{B}^{[1]} = [\tilde{b}, \tilde{b}, ..., \tilde{b}]$，大小为$m \times \hat{n}_1$。

接圆回归的前向计算为

$$\begin{aligned} Z^{[0]} &= \min\left(W^T X^{[0]} + B^{[0]}, -W^T X^{[0]} + \tilde{B}^{[0]}\right), \\ \bar{z}^{[0]} &= \text{mean}(Z^{[0]}), \\ a^{[0]} &= \sigma(\bar{z}^{[0]}), \end{aligned}$$

负样本集上的损失函数为

$$H_1^{[0]}(W,b,\tilde{b}) = -\frac{n_1}{n_0 \hat{n}}\sum_{i=1}^{n_0}\ln(1-a_i^{[0]}).$$

对正样本做类似的向前计算为

$$\begin{aligned} Z^{[1]} &= \min\left(W^T X^{[1]} + B^{[1]}, -W^T X^{[1]} + \tilde{B}^{[1]}\right), \\ \bar{z}^{[1]} &= \text{mean}(Z^{[1]}), \\ a^{[1]} &= \sigma(\bar{z}^{[1]}), \end{aligned}$$

正样本集上的损失函数为

$$H_1^{[1]}(W, b, \tilde{b}) = -\frac{1}{\hat{n}} \sum_{i=1}^{n_1} \ln a_i^{[1]}.$$

为求偏导，令

$$F^{[0]} = \left(W^T X^{[0]} + B^{[0]} \leq -W^T X^{[0]} + \tilde{B}^{[0]}\right), \quad \tilde{F}^{[0]} = 1 - F^{[0]},$$

易得

$$\begin{aligned} \frac{\partial H_1^{[0]}(W, b, \tilde{b})}{\partial b} &= \frac{n_1}{n_0 m \hat{n}} F^{[0]} a^{[0]T} \\ \frac{\partial H_1^{[0]}(W, b, \tilde{b})}{\partial \tilde{b}} &= \frac{n_1}{n_0 m \hat{n}} \tilde{F}^{[0]} a^{[0]T} \\ \frac{\partial H_1^{[0]}(W, b, \tilde{b})}{\partial W} &= \frac{n_1}{n_0 m \hat{n}} X^{[0]} \,\text{diag}(a^{[0]})(F^{[0]} - \tilde{F}^{[0]})^T \end{aligned}$$

这里的$\text{diag}(a^{[0]})$将向量转为对角线矩阵，此处只是为数学表达方便，实际计算机程序代码中不要转成稠密矩阵，否则会使计算量增加d倍。

为求偏导，令

$$F^{[1]} = \left(W^T X^{[1]} + B^{[1]} \leq -W^T X^{[1]} + \tilde{B}^{[1]}\right), \quad \tilde{F}^{[1]} = 1 - F^{[1]}.$$

易得

$$\begin{aligned} \frac{\partial H_1^{[1]}(W, b, \tilde{b})}{\partial b} &= \frac{1}{m \hat{n}} F^{[1]} (a^{[1]T} - 1), \\ \frac{\partial H_1^{[1]}(W, b, \tilde{b})}{\partial \tilde{b}} &= \frac{1}{m \hat{n}} \tilde{F}^{[1]} (a^{[1]T} - 1), \\ \frac{\partial H_1^{[1]}(W, b, \tilde{b})}{\partial W} &= \frac{1}{m \hat{n}} X^{[1]} \,\text{diag}(a^{[1]} - 1)(F^{[1]} - \tilde{F}^{[1]})^T. \end{aligned}$$

7. 实验

本节给出接圆回归的直观几何解释，在公开数据集上对比接圆回归与FM的性能。FM的训练、预测采用xlearn，接圆回归的训练、预测采用自编Matlab程序。

实际业务的数据集维数通常很大，难以直观显示。为了观察FM和接圆回归的样子，这里设计3个二维数据集：星环集、双堆集、双月集，即子图1def 中的散点，蓝点是正样本，红点是负样本。

取隐向量数为2，在这3个数据集上分别训练FM，得到FM在这3个数据集上的预测值$\hat{y}$，见图1。第一行的3个子图是立体图，竖轴是预测值$\hat{y}$，颜色越接近黄色预测值越接近1，颜色越接近蓝色预测值越接近0。第二行的3个子图是立体图的俯视图，颜色含义与立体图一致。在星环集上，FM无法有效分隔正负样本，如子图1a所示，只有几个点上的预测值大于0.6或小于0.4；如子图1d所示，星

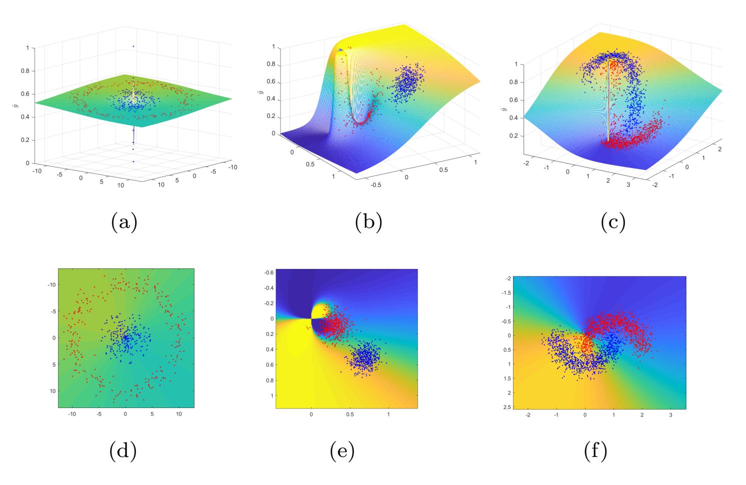


图 1: FM在星环集(ad)、双堆集(be)、双月集(cf)上的效果。

表 1: FM和接圆回归在Avazu 集上的表现。对FM，k是隐向量数量，λ是SGD迭代步长，η是正则化系统；对接圆回归，m是接圆归的折面数，λ是SGD迭代步长，η_1是L2正则化系数，η_2是楔子正则化系数。

模型	参数	验证集AUC
FM	k=16, λ=0.2, $\eta = 0.002$	0.7577
FM	k=32, λ=0.2, $\eta = 0.002$	0.7583
FM	k=64, λ=0.2, $\eta = 0.002$	0.7582
接圆回归	m=16, λ=16,η_1=0, η_2=0.001	0.7607
接圆回归	m=32, λ=16,η_1=0, η_2=0.001	0.7637
接圆回归	m=64, λ=16,η_1=0, η_2=0.001	0.7658

环集所在的2维空间中，左上角的预测值稍大，右下角的预测值稍小，没有突出中心的圆形正类。如子图1be所示，在双堆集上，虽然FM在验证集上的AUC高达0.977，能很好地区分正负样本，但是整个样本空间的预测值曲面不是Lipschitz连续，在某些位置大起大落。在双月集上，如子图1cf所示，FM也在一些区域变化剧烈。

在3个数据集上训练接圆回归，使用3个折面，效果见图2。子图2afk是与式(3.1)对应的单个折面，除了折痕一条线之外，折面上的任意点处梯度均不为零，考虑到折痕的测度为0，因此可以说*折面上的梯度几乎处处不为零*。子图2bgl 均为3个折面放在一起的效果，特别注意，3个折面的折痕都有交于一点。子图2chm均为3个折面叠加后得到的锥面，折痕交点成为锥面的顶点。将锥面进行Sigmoid变换得到子图2din，其俯视图是子图2ejo。这15张子图直观反映了接圆回归的设计目标：*接圆回归用超平面围出一个封闭凸多面体*。对星环集、双堆集来说，都存在一个凸多边形使得正样本全部落在多边形内且负样本全部落在多边形外。但对双月集来说，不存在这个样的凸多边形，因此能在子图2o中看出接圆回归没能将正负样本很好地区分开。

图2显示，接圆回归拟合的曲面Lipschitz连续，变化平缓。这点优于FM。

Avazu集和Criteo集都是著名的点击预测数据集，实验所用数据来自LIBSVM官网[1)] 。Avazu集特征数量99 9999，训练集avazu-app.tr.bz2样本数量1264 2186，验证集avazu-app.val.bz2样本数量195 3951。Criteo集特征数量99 9990，将数据文件criteo.kaggle2014.svm.tar.gz随机划分为训练集和验证集，训练集样本数量1264 2186，验证集样本数量195 3951。表1和表2 显示，在这2个数据集上，接圆回归

[1)] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

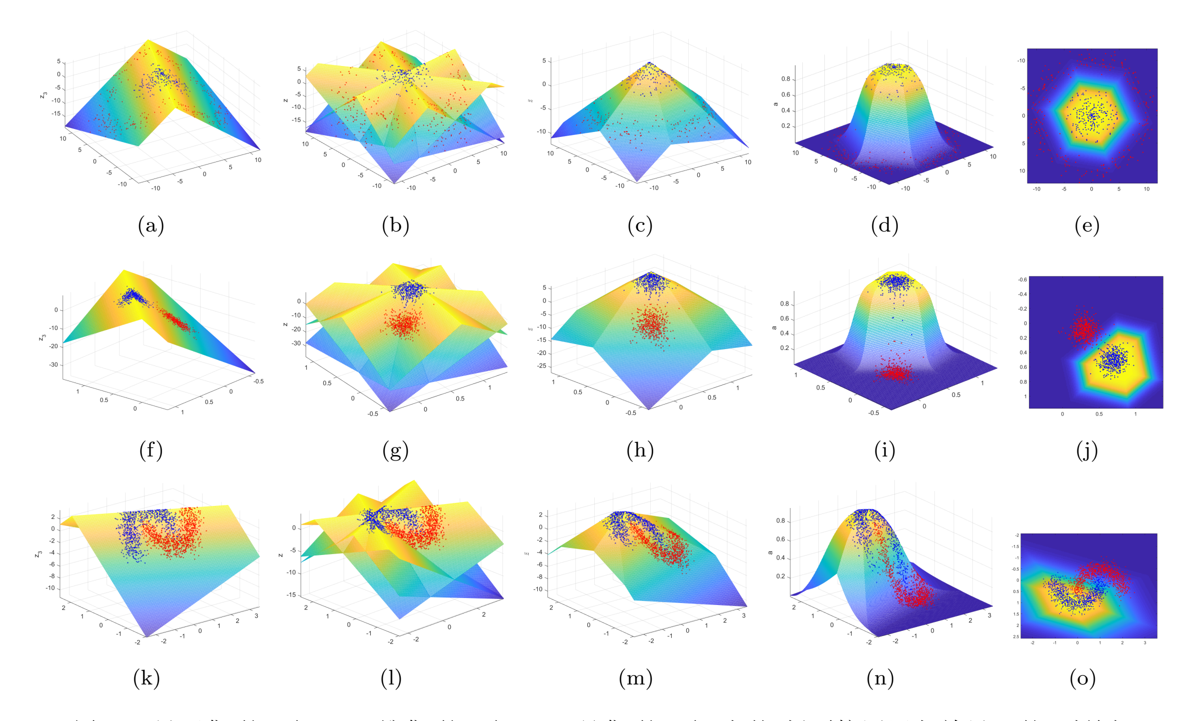


图 2: 星环集(第一行)、双堆集(第二行)、双月集(第三行)上的3折面接圆回归效果。第1列是与式(3.1)对应的1个折面，第2列是与式(3.1)对应的3个折面放在一起；第3列是与式(3.2)对应的z；第4列是与式(3.3)对应的分数a；第5列是第4行的俯视图。

表 2: FM和接圆回归在Criteo 集上的表现。

模型	参数	验证集AUC
FM	k=4, λ=0.2, $\eta = 0.002$	0.7934
FM	k=16, λ=0.2, $\eta = 0.002$	0.7941
FM	k=64, λ=0.2, $\eta = 0.002$	0.7947
接圆回归	m=4, λ=1,η_1=0, η_2=0.001	0.7732
接圆回归	m=16, λ=8,η_1=0, η_2=0.001	0.7951

的AUC略优于FM。

从本节的实验中可以看出接圆回归的几个优点：有直观的几何解释，拟合的曲面变化平缓（Lipschitz 连续），任意初值都收敛到全局最优点，AUC和损失函数值单调增、减，AUC随折面数量增加而增加。

8. 讨论和总结

定义 1. *对数据集D，其中的正样本集记为D_1，负样本集记为D_0。如果都存在一个封闭凸区域，使得D全部落在多边形内且D_0全部落在多边形外，那么称D是凸可分的。*

接圆回归的直觉设计目标是

猜想 1. *在凸可分集D上，接圆回归的目标函数$H(W,b,\tilde{b})$几乎处处严格凸。*
如果猜想1成立，那么接圆回归训练时，对任意初值，式(3.8)都能一次收敛到全局最优解，克服了FM的缺点。

由定义1可知，星环集、双堆集都是凸可分的，双月集不是凸可分的，微博集、Avazu集和Criteo集不确定是否凸可分。但是，在实际训练中，接圆回归在这6个数据集上都是对任意初始值一次性收敛到全局最优解，这意味着猜想很可能是成立的，甚至在更宽松的条件下成立。

实验不能代替证明，猜想1还需要严格的数学证明。

在深度CTR模型中，接圆回归有望接替逻辑回归的位置，例如将Deep & Cross最后一层的逻辑回归更换为接圆回归。楔子正则也可以应用到各种深度学习模型当中。

参考文献

[1] 周志华, 机器学习, p57-60，清华大学出版社，2016.4

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research 9(2008), 1871-1874

[3] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, Zhe Wang. Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction. arXiv:1704.05194. 18 Apr 2017.

[4] Stemffen Rendle. Factorization Machines, in Proceeding of the 10th IEEE International Conference on Data Mining (ICDM 2010), Sydney, Australia.

[5] Stemffen Rendle. Factorization Machines with libFM, in ACM Trans. Intell. Syst. Technol., 3(3), May, 2012.

[6] Stemffen Rendle. Learning Recommender System with Adaptive Regularization. Proceedings of the fifth ACM international conference on Web search and data mining. ACM, 2012: 133-142.

[7] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, Chih-Jen Lin. Field-aware Factorization Machines for CTR Prediction. RecSys '16 Proceedings of the 10th ACM Conference on Recommender Systems Pages 43-50. Boston, Massachusetts, USA —September 15 - 19, 2016.

[8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, et al. Wide & Deep Learning for Recommender Systems. arXiv:1606.07792. 24 Jun 2016.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. arXiv:1804.04950. 16 May 2018.

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He. DeepFM: An End-to-End Wide & Deep Learning Framework for CTR Prediction. arXiv:1703.04247. 13 Mar 2017.

[11] Xiangnan He, Tat-Seng Chua. Neural Factorization Machines for Sparse Predictive Analytics. arXiv:1708.05027. 16 Aug 2017.

[12] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, et al. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. arXiv:1708.04617. 15 Aug 2017

[13] Ruoxi Wang, Bin Fu, Gang Fu, Mingliang Wang. Deep & Cross Network for Ad Click Predictions. arXiv:1708.05123. 17 Aug 2017.

[14] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, et al. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. arXiv:1803.05170. 30 May 2018.

[15] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan, et al. Deep Interest Evolution Network for Click-Through Rate Prediction. arXiv:1706.06978. 13 Sep 2018.